

Code-Snippets für die Kommandozeile und Batchdateien

[Cmd-Referenz ss64.com](#)

[Cmd-Referenz \(Microsoft\)](#)

[Cmd-Hilfe Win 7](#)

[Cmd-Hilfe Win 7 als Pdf](#)

[Cmd-Hilfe Win 10](#)

[Cmd-Hilfe Win 10 als Pdf](#)

[NT-Kurs Technikerschule](#)

[Batchscripting Wikibooks englisch](#)

[Batchscripting Wikibooks deutsch](#)

[Batch-Infos](#)

1. einfache Befehle

md (make directory) einen Ordner erstellen
rd (remove directory) einen Ordner löschen
dir Inhaltsverzeichnis auflisten
d: wechselt ins Laufwerk D:
cd (change directory) in einen Ordner wechseln
cd.. einen Ordner höher wechseln
**cd ** direkt ins Root-Verzeichnis wechseln
 * Platzhalter für verschiedene Buchstabenfolgen: **dir *.txt** bedeutet: liste alle Dateien mit der Endung .txt auf
copy eine Datei kopieren: **copy Quelle Ziel**
 > Umleitung der Ausgabe eines Befehles: **dir *.txt > c:\inhalt.txt** schreibt alle Txt-Dateien nach C: \inhalt.txt
 >> **dir *.txt >> c:\inhalt.txt** schreibt alle Txt-Dateien nach C: \inhalt.txt. Existiert die Datei bereits, wird das neue Ergebnis in der Datei dazugeschrieben.

Command-Line Syntax Key

Notation	
Text without brackets or braces	Items you must type as shown
<Text inside angle brackets>	Placeholder for which you must supply a value
[Text inside square brackets]	Optional items
{Text inside braces}	Set of required items; choose one
Vertical bar ()	Separator for mutually exclusive items; choose one
Ellipsis (...)	Items that can be repeated

2. Explorer und Browser starten

```
start C:\users\anwender\  
rem start D:\Html_Allmann\JavaScript-Codes_w3schools  
rem start C:\Windows\  
start firefox.exe  
exit
```

3. eine Kommandozeile, den Explorer und die Ordneroptionen starten

```
start "Papa-Konsole" /d C:\Users\ari\Desktop /max cmd.exe  
/t:8e  
start D:\  
start control folders  
exit
```

4. pause

Beim PAUSE-Befehl werden weitere Angaben in derselben Zeile ignoriert. Allerdings werden eventuelle Umleitungszeichen <|> abgearbeitet, was selten Sinn macht - bis auf diesen Fall: **PAUSE > NUL** Damit wird die Ausgabe der Standardaufforderung ins Nichts umgeleitet, falls man etwas anderes per ECHO formuliert hat.

```
@echo off
echo Hier schreibe ich jetzt ganz viel Text.
echo Hier kann ich z.B. hinschreiben, dass ich jemanden mag.
echo.
echo Aber den Text sieht man gleich sowieso nicht mehr ... Hihi!
cls
echo War irgendetwas?
pause>nul
rem Mit pause>nul wird der Text "Drücken Sie eine beliebige Taste ..." nicht angezeigt.
```

```
cmd C:\Windows\system32\cmd.exe
War irgendetwas?
```

5. copy

Der Quellpfad kann absolut (`c:\users\allmann\desktop\test.txt`) oder relativ (`test.txt`) angegeben werden. Bei relativer Angabe wird die Datei im momentanen Verzeichnis gesucht.

`copy c:\users\benutzername\test.docx d:\textverarbeitung\neu-test.docx` kopiert die Datei `test.docx` aus dem Ordner `c:\users\benutzername` in das Laufwerk D: und dort in den Ordner `textverarbeitung` und benennt die Datei in `neu-test.docx` um.

`copy raubvoegel.odt c:\voegel` kopiert die Datei vom aktuellen Laufwerk nach C: und legt dort den Ordner `voegel` an, falls dieser noch nicht existiert.

`copy *.txt kombiniert.txt` verbindet alle Dateien mit der Endung `txt` im momentanen Verzeichnis in einer Datei mit dem Namen `kombiniert.txt`.

6. Datum und Zeit

```
@echo off
date /t
rem Zeigt nur das Datum und verlangt keine Datumseingabe.
echo Heute ist der %date%.
rem Das Datum wird als Variable angezeigt.
echo Dieses Jahr ist das Jahr %date:~6,4%.
rem Die ersten 6 Zeichen werden ausgeblendet, dann werden 4
Zeichen angezeigt.
```

Anzeige in der Cmd:

```
22.01.2018
Heute ist der 22.01.2018.
Dieses Jahr ist das Jahr 2018.
```

```
@echo off
rem start d:\
time /t
echo Die aktuelle Zeit ist %time%!
echo Kuerzer: %time:~0,6%
echo Das ist mein aktuelles Verzeichnis: %cd%
```

7. Computer herunterfahren / abmelden

`rem Dieses Programm fährt den Rechner herunter / meldet ihn ab.`

`rem Die mit "call" aufgerufenen bat-Dateien müssen natürlich zur Verfügung stehen.`

```
@echo off
:Anfang
echo.
echo Bitte einen der folgende Buchstaben zum Beenden des
Rechners eingeben und mit ENTER bestaetigen:
echo.
echo.
echo.
echo N = Neustart eingeben
echo.
echo H = schnelles Herunterfahren
echo.
```

```

echo A = Abmelden
echo.
echo Q = Abbrechen der Kommandozeile
echo.
echo.
set /p buchstabe=Bitte n, h, a oder q eingeben:
if %buchstabe%==q goto Ende
if %buchstabe%==n goto neustart
if %buchstabe%==h goto herunter
if %buchstabe%==a goto abmelden
:neustart
shutdown /r /t 0
exit
:herunter
shutdown /p /f
exit
:abmelden
shutdown /l
exit
rem Falls eine falsche Angabe eingetippt wird:
echo Falsche Eingabe %buchstabe% !
rem Dann wartet das Programm auf eine Bestätigung:
pause
rem Und sprint mit goto wieder an den Anfang:
goto Anfang
rem Sprungmarke fuer das Beenden:
:Ende
exit

```

8. cd

Mit `cd \` wechselt man sofort in den Hauptordner des Laufwerkes. So navigiert man z.B aus `C:\users\benutzer` mit dem Befehl `cd \windows\system32` sofort in diesen Ordner, ohne mit verschiedenen `cd`-Befehlen dorthin wechseln zu müssen. Der Befehl `cd /d %~dp0` wechselt in den Ordner des jeweiligen Batchscripts - sinnvoll bei Doppelklicks.

Einen Pfad in eine Variable schreiben:

Mit `set aktuellerpfad=%cd%` wird das aktuelle (= gerade verwendete) Verzeichnis in die Variable `%aktuellerpfad%` geschrieben. Damit lässt sich am Skriptende mit `cd %aktuellerpfad%` wieder schnell zurück ins Startverzeichnis wechseln.

9. Sprungziel des GOTO-Befehls

Im GOTO-Befehl darf der angegebenen Marke ein Doppelpunkt vorangestellt werden:

```
Goto :sprungmarkenname
Goto sprungmarkenname
```

ist ebenfalls gültig

Die System-Sprungmarke `:EOF` ist vordefiniert und bewirkt das Verlassen der Batchdatei bzw. der aktuellen Unterroutine, wenn vorher eine Unterroutine mit `CALL` aufgerufen wurde.

10. for

Prinzipieller Aufbau: `for... in... do...`

alle Textdateien im aktuellen Verzeichnis anschauen

```
@ECHO OFF
FOR %a IN (*.txt) DO type %a
```

Bei FOR immer Variablen mit einem Buchstaben verwenden. In Batch-Dateien muss ein % verwendet werden - auf der Kommandozeile zwei %%.

alle Dateien im Temp-Verzeichnis anschauen

```
@echo off
for /r %temp% %%f in (*.*) do (
echo %%f
)
```

Es werden nur Dateien aber keine Verzeichnisse angezeigt. Um Verzeichnisse

anzuzeigen, muss for /d verwendet werden. Der Parameter /r bewirkt, dass alle Unterverzeichnisse mit einbezogen werden (Rekursive Schleife).

Zählschleifen

Syntax: `for /L {Variable} IN (Startzahl, Schrittweite, Endzahl) DO (Aktion)`

REM Schreibe einen Text fuenf Mal

@echo off

for /L %%N IN (1, 1, 5) DO echo Nummer %%N

pause

Anzeige in der cmd:

Nummer 1

Nummer 2

Nummer 3

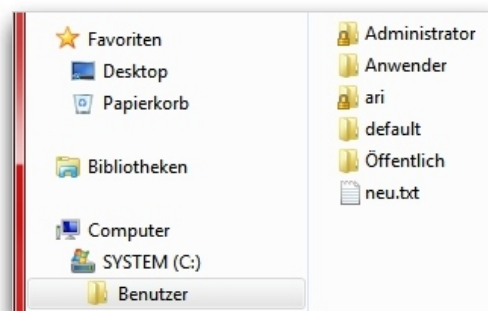
Nummer 4

Nummer 5

Drücken Sie eine beliebige Taste ...

Anzeige aller Dateien im aktuellen Ordner mit der entsprechenden Endung

`for %i in (*.txt) do @echo %i`



```
C:\Users>for %i in (*.txt) do @echo %i
neu.txt
```

Auflistung von Dateien um Umleitung in eine Textdatei

`(for %i in ("C:\Windows\system32*.exe") do @echo %i) > exe.txt`

Alle exe-Dateien des system32-Ordners werden in die Datei exe.txt geschrieben. Ohne Klammer würde nur die allerletzte exe-Datei geschrieben werden, da bei jedem for-Durchgang eine Datei geschrieben wird und dann durch die nächste überschrieben wird.

Alle Ordner inklusive aller Unterordner anzeigen (ausgehend vom aktuellen Ordner)

`for /r /d %i in (*) do @echo %i`

```
D:\software>for /d /r %i in (*) do @echo %i
D:\software\2017
D:\software\MediaplayerClassic
D:\software\notepad-Versionen
D:\software\paint.net.4.0.12.install
D:\software\Ps 3.0
D:\software\Videopad
D:\software\WindowsEnablerv1.1
D:\software\2017\PS-help
D:\software\2017\PS-help für WMF 5.0
D:\software\2017\WMF 4.0
D:\software\2017\WMF 5.0
D:\software\2017\PS-help für WMF 5.0\englische_kultur
D:\software\MediaplayerClassic\MPC-HC.1.7.8.x64
D:\software\MediaplayerClassic\MPC-HC.1.7.8.x64\MPC-HC.1.7.8.x64
D:\software\MediaplayerClassic\MPC-HC.1.7.8.x64\MPC-HC.1.7.8.x64\Lang
D:\software\MediaplayerClassic\MPC-HC.1.7.8.x64\MPC-HC.1.7.8.x64\LAVFilters64
D:\software\MediaplayerClassic\MPC-HC.1.7.8.x64\MPC-HC.1.7.8.x64\Shaders
D:\software\notepad-Versionen\npp.6.7.4.bin
```

Jeder Ordner wird mit vollem Pfad dargestellt.

11. **if**

Prüft, ob eine Bedingung erfüllt oder nicht erfüllt ist.

Existenz einer Datei prüfen

Ein Beispiel, das eine Datei nur dann kopiert, wenn sie existiert und sonst eine Fehlermeldung ausgibt, wäre etwa:

```
@ECHO OFF
IF NOT EXIST datei.txt GOTO fehler
COPY datei.txt c:\verz1
GOTO ende
:fehler
ECHO Datei datei.txt nicht gefunden !
:ende
```

Tipp: Um zu überprüfen, ob ein Verzeichnis existiert, kann man einfach die Existenz der Datei NULL in diesem Verzeichnis abfragen: Existiert sie, so existiert auch das Verzeichnis.

Zeichenkettenvergleich

Syntax: `IF [NOT] "Zeichenkette1"=="Zeichenkette2" Befehl`

Hier können zwei Zeichenketten miteinander verglichen werden, beide sollten immer in doppelten Anführungszeichen stehen. Als Zeichenketten können sowohl eingetippte Wörter, Umgebungsvariablen als auch Kommandozeilenparameter benutzt werden. Ein Beispiel, das überprüft, ob ein Kommandozeilenparameter eingegeben wurde:

```
@ECHO OFF
IF "%1"==" " GOTO fehler
...
GOTO ende
:fehler
ECHO Keine Parameter eingegeben...
:ende
```

12. **Programmierhilfen**

Um einen anderen Editor als Notepad standardmäßig verwenden zu können, muss in der Registry folgende Änderung gemacht werden: In

`HKEY_CLASSES_ROOT\batfile\shell\edit\command` den Standardwert auf den Pfad eines besseren Editors ändern, z.B `D:\software\Notepad-Versionen\npp.6.7.4.bin\notepad++.exe %1`

13. **Variablen**

Typen von Umgebungsvariablen

Windows unterscheidet zwischen **Benutzervariablen** (User Environment), **Systemvariablen** (Machine Environment) und **temporären Umgebungsvariablen** (Volatile Environment), die alle sowohl gelesen als auch verändert werden können. Darüber hinaus gibt es noch **interne Variablen** (z.B. `%ComputerName%`, `%SystemRoot%`, etc.), die zwar gelesen aber nicht (direkt) verändert werden können.

Falls mehrere gleichnamige Variablen definiert wurden, haben temporäre Variablen Vorrang vor Benutzervariablen und diese wiederum Vorrang vor Systemvariablen (d.h. wenn die Variable TEST im User Environment und im Machine Environment existiert, wird der Wert der Benutzervariablen TEST angezeigt). Ausnahme: Bei der PATH-Variable werden ggf. die Werte mehrerer vorhandener Variablen automatisch durch Semikolon getrennt hintereinander gehängt.

Spezielle Variablen

Wenn die Befehlsweiterungen aktiviert sind, gibt es mehrere dynamische Umgebungsvariablen, die zwar erweiterungsfähig sind, aber nicht mit SET angezeigt werden. Diese Variablenwerte werden bei jeder Werterweiterung der Variable dynamisch berechnet. Falls eine Variable mit einem dieser Namen definiert wird, überschreibt diese Definition die unten stehende dynamische Definition:

<code>%CD%</code>	erweitert zum momentanen Verzeichnisnamen
-------------------	---

%DATE%	erweitert zum aktuellen Datum im HH:MM:SS.mm-Format (wie der DATE-Befehl)
%TIME%	erweitert zur aktuellen Zeit unter Verwendung desselben Formats wie der TIME-Befehl
%RANDOM%	erweitert zu einer zufällig gewählten Dezimalzahl zwischen 0 und 32767
%ERRORLEVEL%	erweitert zum ERRORLEVEL-Wert des letzten Befehles
%CMDEXTVERSION%	erweitert zur Versionsnummer der momentan verwendeten Kommandozeile
%CMDCMDLINE%	erweitert zur ursprünglichen Befehlszeile, als die Kommandozeile gestartet wurde
%HIGHESTNUMANODENUMBER%	erweitert zu der höchsten NUMA-Knotennummer auf diesem Computer

14. Syntax

@

Schaltet die Ausgabe der Befehlszeile auf dem Bildschirm für den aktuellen Befehl aus.

echo Diese Zeile wird mit Befehlszeile ausgeführt...

@echo und diese ohne!

Anzeige in der cmd.exe:

C:\>echo Diese Zeile wird mit Befehlszeile ausgeführt...

Diese Zeile wird mit Befehlszeile ausgeführt...

und diese ohne!

!

Dient zur verzögerten Übersetzung von Variablen. Bewirkt, dass die Variable nicht zur Kompilierzeit sondern erst zur Laufzeit übersetzt wird. Setzt die Verwendung von SETLOCAL zur Aktivierung von verzögerter Übersetzung voraus.

```

C:\Users\Anwender\Desktop\new 0.cmd
Datei Bearbeiten Suchen Ansicht Ko
new 0.cmd
1 @echo off
2 setlocal EnableDelayedExpansion
3 set _var=eins
4
5 set _var=zwei
6 echo !_var%! !_var! text
7 pause
8 exit
9 Resultat: zwei zwei text

```

Kommandozeilen-Parameter

a) Batch-Dateien können Kommandozeilenparameter bearbeiten

Diese Parameter werden innerhalb der Datei mit %1 bis %9 bezeichnet. Um z.B. eine Batch-Datei zu erstellen, die den Inhalt eines Datenträgers in ein bestimmtes Verzeichnis kopiert, könnte man schreiben:

@Echo off

Echo Bitte den Datenträger anschließen!

Pause

```
Copy D:\*.* %1
```

```
Echo Fertig!
```

Würden man diese Datei dcopy.cmd nennen und sie folgendermaßen aufrufen:

```
dcopy C:\VERZ1
```

so stünde in der Zeile mit dem Kopierbefehl jetzt statt des %1 der angegebene Parameter C:\VERZ1 und der Befehl würde lauten:

```
Copy D:\*.* C:\VERZ1
```

Es sind genau neun Parameter mit dieser Methode ansprechbar. Der Parameter %0 hat eine Sonderbedeutung: er enthält den Namen der Batch-Datei.

b) Befehlsparameter %1

In Cmd-Abläufen werden oft variable Angaben gebraucht, die erst mit dem jeweiligen Aufruf festgelegt werden sollen. Dazu werden in der Cmd-Datei "Platzhalter" eingebaut. Beim Start einer Cmd-Datei werden dann die aktuellen Werte einfach zusätzlich in der Befehlszeile angegeben (Befehlsparameter). Platzhalter bestehen aus einem Prozentzeichen mit der laufenden Nummer des Befehlsparameters, also %1 %9 (nur eine Ziffer möglich). Ein Beispiel mit der Datei EDR.cmd:

```
attrib -R %1
```

```
EDIT %1
```

```
attrib +R %1
```

Aufruf: EDR ANY.TXT

Hier wird vor dem Aufruf des Editors (EDIT) der Schreibschutz der angegebenen Datei entfernt und anschließend wieder eingeschaltet. Die CMD ersetzt die Platzhalter jeweils, bevor eine Zeile interpretiert wird, durch den entsprechenden Befehlsparameter. Auf diese Weise kann praktisch alles in einem BAT-Ablauf variabel gemacht werden, selbst Befehle und Programm-Aufrufe.

c) Trennzeichen, Sonderzeichen

Zur Trennung von Befehlsparametern können auch Komma oder Semikolon verwendet werden. Sie werden praktisch durch Leerzeichen ersetzt. Auch das Gleich-Zeichen "=" wird auf die selbe Weise behandelt. Leere Parameter können auf diese Weise NICHT übergeben werden:

```
XXX.BAT eins,,vier
```

ergibt:

```
%1=eins
```

```
%2=vier
```

Übergabe eines Parameters, in dem Leerzeichen enthalten sein sollen:

```
Datei.cmd "Eigene Dateien"
```

ergibt: %1="Eigene Dateien"

Zum Ausprobieren empfiehlt es sich, eine TEST.BAT zu schreiben, die einfach 9 ECHO-Befehle enthält:

```
@echo off
```

```
ECHO.%1
```

```
ECHO.%2
```

```
(etc.)
```

Dann kann diese BAT-Datei mit allen möglichen Kombinationen getestet werden. Der Punkt nach ECHO sorgt dafür, daß bei leeren Parametern nicht die Meldung "ECHO ist ausgeschaltet" kommt.

d) Parameter %0

Mit %0 kann man im Batch-Ablauf den Namen der Cmd-Datei ansprechen, genauer gesagt, den Befehl, so wie er beim Aufruf angegeben wurde.

e) Shift-Befehl

SHIFT verschiebt die Parameter-Liste nach links, d.h. %0 fällt hinaus, der bisherige %1 wird zu %0, %2 wird zu %1 und so weiter. Damit kann auch der zehnte Parameter erreicht werden (jetzt %9). SHIFT kann bei Bedarf wiederholt werden.

f) Beispiele

```
@echo off
```

```
REM Diese Batchdatei ruft eine eigene Sprungmarke auf
```

```
echo Vor dem Sprung
```

```
call :sprungmarke einParameterderSprungmarke
```

```
echo Nach dem Sprung
```

```
pause
```

```
goto end
```

```
:sprungmarke
```

```
echo Sprungmarke aufgerufen und Parameter %1 uebergeben!
```

```
goto :eof
```

```
:end
```

```
exit
```


Die Ausgabe:

Vor dem Sprung

Sprungmarke aufgerufen und Parameter einParameterderSprungmarke uebergeben!

Nach dem Sprung

Anmerkung: Mit `goto :eof` springt die Prozedur nach dem Aufruf von `:sprungmarke` zurück an das Ende des `call`-Befehles und führt das Programm weiter. Wird der Programmaufruf nur in eine `Cmd` kopiert und nicht als Batchdatei aufgerufen, zeigt die Fehlermeldung, dass das Sprungziel nicht innerhalb dieser `Cmd`-Instanz gefunden werden kann. Der Programmablauf ab der Sprungmarke wird nicht ausgeführt und das Programm kann auch nicht das Sprungziel `:end` finden und bleibt deshalb mit der Pause hängen

```
Administrator: Windows-Befehlsprozessor
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\users\ari\Desktop>@echo off
REM Diese Batchdatei ruft eine eigene Sprungmarke auf
echo Vor dem Sprung
Vor dem Sprung
call :sprungmarke einParameterderSprungmarke
Ungültiger Versuch, ein Sprungziel außerhalb einer Batchdatei aufzurufen.
echo Nach dem Sprung
Nach dem Sprung
pause
Drücken Sie eine beliebige Taste . . .
```

```
sprungmarke.bat
1 @echo off
2 REM Diese Batchdatei ruft eine eigene Sprungmarke auf
3 echo Vor dem Sprung
4 call :sprungmarke EINS ZWEI
5 echo Nach dem Sprung
6 pause
7 goto end
8
9 :sprungmarke
10 echo Sprungmarke aufgerufen und Parameter %2 und %1 uebergeben!
11 goto :eof
12 :end
13 exit
```

```
C:\Windows\system32\cmd.exe
Vor dem Sprung
Sprungmarke aufgerufen und Parameter ZWEI und EINS uebergeben!
Nach dem Sprung
Drücken Sie eine beliebige Taste . . .
```

```
Pfad dp0.bat
1 @echo off
2 echo Dieses Script wurde im Pfad %~dp0 gestartet.
3 pause
```

```
@echo off
echo Dieses Script wurde im Pfad %~dp0 gestartet.
pause
exit
```

```
C:\Windows\system32\cmd.exe
Dieses Script wurde im Pfad D:\Verschiedenes\Kommandozeile\batch\Scripts\ gestartet.
Drücken Sie eine beliebige Taste . . .
```

In diesem Script starten die Ordneroptionen und Notepad auch ohne den Befehl `start`:

```
@echo off
echo Wir sind in: %~dp0
```



```
control folders  
notepad
```

A. Allmann, NMS Rennweg am Katschberg (Kärnten)